

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2002-182953

(P2002-182953A)

(43)公開日 平成14年6月28日(2002.6.28)

(51)Int.Cl. ⁷	識別記号	F I	テーマコード [*] (参考)
G 0 6 F 12/00	5 0 1	G 0 6 F 12/00	5 0 1 H 5 B 0 6 5
	5 1 4		5 1 4 M 5 B 0 8 2
3/06	3 0 2	3/06	3 0 2 A 5 D 1 1 0
G 1 1 B 27/00		G 1 1 B 27/00	A

審査請求 未請求 請求項の数 2 O L (全 13 頁)

(21)出願番号 特願2000-383124(P2000-383124)

(22)出願日 平成12年12月12日(2000.12.12)

(71)出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(71)出願人 000233055

日立ソフトウェアエンジニアリング株式会
社

神奈川県横浜市中区尾上町6丁目81番地

(72)発明者 西村 祐二

神奈川県横浜市戸塚区戸塚町5030番地 株
式会社日立製作所ソフトウェア開発本部内

(74)代理人 100075096

弁理士 作田 康夫

最終頁に続く

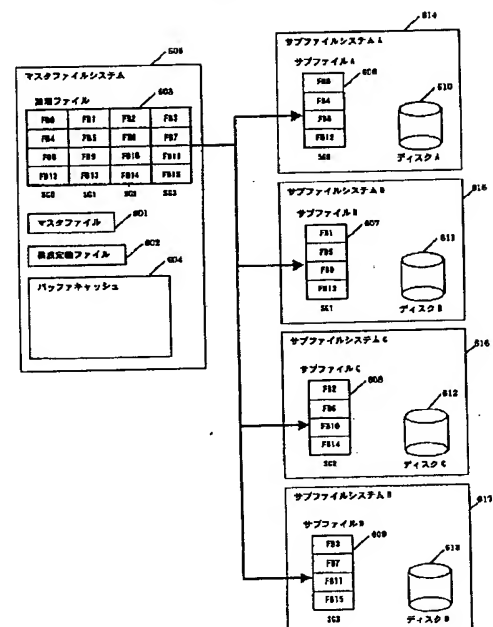
(54)【発明の名称】 分散ファイル管理方法

(57)【要約】

【課題】 ストライピングファイル機構により複数の論理ファイルをディスク装置に分散配置する方法において、I/O要求時の並列ディスク装置アクセスの高速化を実現する。

【解決手段】 論理ファイル603をサブファイルシステム614~617に分散配置する方法で、同一ファイルのファイルブロックFBO~15がディスク装置610~613の一括した領域に配置するためにファイルブロックの集合であるストライドグループSG0~SG3を生成して、各サブファイルシステムにストライドグループ単位で一括して書き込む。また、ファイルブロックアクセス時にマスタファイルシステム605のバッファキャッシュ604に連続するファイルブロックFB4~15を先読みし、サブファイルシステムへの読み出し要求の回数を削減し、ファイルアクセスの更なる高速化を実現する。

図 6



【特許請求の範囲】

【請求項 1】 複数ディスク装置からなるファイルシステム上で、論理ファイルを一定のサイズの複数ファイルブロックに分割し、これらのファイルブロックを複数ディスク装置に分散して配置することで、ファイル読み出し時に複数ディスク装置に並行してアクセスすることが可能なファイルシステムにおいて、ファイルブロックを分散配置することにより、物理的に一ディスク装置内で、同一ファイルのファイルブロックが不連続領域に配置されていたものを、ファイルブロックに管理情報を付与することで連続配置し、一回の I/O 操作で読み出し／書き込みができることを特徴とする分散ファイル管理方法。

【請求項 2】 請求項 1 項の分散ファイル管理方法において、ファイルシステム上に論理ファイルの管理情報を維持し、論理ファイルのファイルブロックに読み出し要求が生じた際、その情報に従い各ディスク装置で次に読み出すファイルブロックを予め読み出して、これらのファイルブロックをファイルシステム上のバッファキャッシュに保存することで、ディスクへの読み出し回数を削減することを特徴とする分散ファイル管理方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、マスタファイルシステムに仮想的に存在する論理ファイルを、複数ディスク装置に分割して保存することで、並行して複数ディスク装置のファイルブロックにアクセス出来るようにするストライピングファイル機構と呼ばれるファイル運用方法に関するものである。

【0002】

【従来の技術】 従来より、特開平 9-223049 号公報に記載のような、ストライピングファイル機構による、論理ファイルを複数ディスク装置に分割して配置する方法で、論理ファイルへの I/O 要求時に、複数ディスク装置に並行してアクセスできることは知られており、並列計算機での大容量ファイル管理において、本ファイルアクセス技術は、広く利用されている。

【0003】

【発明が解決しようとする課題】 上記従来技術は、論理ファイルを複数ディスク装置に分散配置することで、並列アクセスを可能にして、大容量ファイルへの読み出し要求の高速化を図るものであったが、ディスク装置への書き込み時、分割されたファイルブロック群に対して、同一ディスク装置へ格納するデータ群という認識をせず、各ファイルブロックを個別にディスク装置に格納するため、ディスク装置上でのファイルブロックの配置にばらつきが生じ、ファイル読み出し時、ディスク装置の回転待ち、シーク（ファイルブロックの検索）の回数が増加するため、性能劣化の原因となっていた。

【0004】 本発明の目的は、ストライピングファイル

機構により論理ファイルを複数ディスク装置に分散配置する方法において、ファイル読み出し時のディスク装置へ並列での読み出しの効率化を図り、大容量ファイルへの読み出し時間の短縮をすることにある。

【0005】

【課題を解決するための手段】 上記目的を達成するために、ストライピングファイル機構による論理ファイルの分散管理方法において、論理ファイルを複数ディスク装置に分散配置するとき、各ディスク装置に配置する同一ファイルのファイルブロックを一括して書き込みしておくことで、ディスク装置からのファイルブロックの読み出し時にシークの回数を削減し、ファイルへのアクセス時間を短縮するものである。また、ファイルのブロックを一括してディスク装置への書き込みをすることで、ファイル読み出し時にマスタファイルシステムのバッファキャッシュに、これからアクセスが発生するファイルブロックの先読みを可能にし、各ディスク装置への読み出し要求の回数を削減するため、ファイルの読み出し時間の短縮化を図るものである。

【0006】

【発明の実施の形態】 以下、本発明の実施の形態について説明する。図 1 は、本発明の一実施例の全体構造であり、101 はマスタファイル、102 は構成定義ファイル、103 は論理ファイル、FB0～FB15 は論理ファイルを一定のサイズに分割したファイルブロック、SG0～SG3 はファイルブロックの集合であるストライドグループ、104 は 101、102、及び 103 を格納するマスタファイルシステム、105～108 はサブファイル A から D、109～112 はディスク装置 A から D、113 から 116 はサブファイル、及びディスク装置を格納するサブファイルシステム A から D を示す。

【0007】 マスタファイルは、論理ファイルのファイルブロックがどのサブファイルに格納しているかの情報を持ち、分散配置された論理ファイルを管理する。構成定義ファイルは、ストライピングファイル機構を使用するうえで、設定すべき情報を定義する。これらの情報には、ストライピングファイル機構が使用するサブファイルシステム、ファイルを分割する際のファイルブロックのサイズが含まれる。論理ファイルは、分割して管理する対象となるファイルで、マスタファイルシステム内に論理上作成される。ファイルブロックは、論理ファイルを一定のサイズに分割したファイルブロックで、サブファイルとして分散管理される最小単位である。ストライドグループは、同一ファイルシステムに分配されるファイルブロックをグループ化したもので、サブファイル分割のときに用いる単位である。マスタファイルシステムは、ストライピングファイル機構を使用するファイルシステムで、論理ファイル、マスタファイルを管理するファイルシステムである。各サブファイルは、サブファイルシステム上の実データが格納されているファイルで

ある。各ディスク装置は、物理的にファイルを格納する装置で、1サブファイルシステムは、1ディスク装置をマウントすることにより、ファイルシステムを作成する。各サブファイルシステムは、サブファイルが格納されているファイルシステムである。

【0008】本発明は、マスタファイルシステムにある論理ファイルを一定のサイズに分割したファイルブロックを、複数のサブファイルシステムのディスク装置に分割して配置し、アプリケーションからのI/O要求をそれぞれのサブファイルが並行して実行することで、ファイルアクセスの高速化をはかるものである。図1で、論理ファイルは、一定のサイズに分割されたファイルブロック0から15で構成している。これらのファイルブロックを複数のディスク装置に配置するマスタファイルシステムは、その構成定義ファイルによりこれらファイルブロックの分配対象となるサブファイルシステムを予め限定している。図1で、サブファイルシステム用にマウントされたディスク装置が4個であるとする。このとき、論理ファイルのファイルブロック0、4、8、12がサブファイルシステムAのディスク装置Aへ、ファイルブロック1、5、9、13がサブファイルシステムBのディスク装置Bへ、ファイルブロック2、6、10、14がサブファイルシステムCのディスク装置Cへ、ファイルブロック3、7、11、15がサブファイルシステムDのディスク装置Dへ分散配置することは、構成定義ファイルから導かれる対象サブファイルシステムとファイルブロック数により決定している。論理ファイルとサブファイルとの関係を管理するマスタファイルは、ファイルブロックをディスク装置に配置するとき、ファイルブロック単位で当該ディスク装置に書き込みをすることなく、最初に同一ディスク装置に分配されるファイルブロックをグループ化する。この一纏にしたファイルブロック群をストライドグループと呼ぶ。ファイルブロック0、4、8、12をストライドグループ0、ファイルブロック1、5、9、13をストライドグループ1、ファイルブロック2、6、10、14をストライドグループ2、ファイルブロック3、7、11、15をストライドグループ3として、各ストライドグループ単位で対象サブファイルシステムのディスク装置へ一括して書き込む。

【0009】図2に、本方法における一ファイル分配処理の詳細を説明する。図2は、本分散ファイル管理方法を実現するために、マスタファイルシステムが論理ファイルの分割時に行う処理について示すものである。S201で、構成定義ファイルより、ストライピングファイル機構に使用可能なサブファイルシステムとその数を、またファイルを分割するファイルブロックのサイズを読み出す。S202で論理ファイルをファイルブロックサイズに分割し、分割したファイルブロックの総数をn個とする。分割したファイルブロックを前から順番にプロ

ック番号0からn-1とする。このとき、最後のファイルブロックが定義されたファイルブロックサイズに満たないときでも、1ファイルブロックとみなし、ブロック番号を定義する。ファイルブロックにブロック番号を定義した後、S203で、ブロック番号をサブファイルシステム数で割った余りが同じファイルブロックを集めてストライドグループにする。また、この余りをストライドグループ番号とする。S204において、S203で生成したストライドグループをストライドグループ番号の順にサブファイルシステムに割り当てる。

【0010】図3では、上記分散ファイル管理方法を用いて、複数の論理ファイルを同一ディスク装置に分配したときのファイルブロックの配置を説明する。図3において、301はマスタファイル、302は構成定義ファイル、303は論理ファイル1、304は論理ファイル2、FB0~FB15は論理ファイル1を一定のサイズに分割したファイルブロック、SG0~SG3は論理ファイル1のファイルブロックの集合であるストライドグループ、FB16~FB27は論理ファイル2を一定のサイズに分割したファイルブロック、Sg0~Sg3は論理ファイル2のファイルブロックの集合であるストライドグループ、305は301、302、303、及び304を格納するマスタファイルシステム、306~309は論理ファイル1を分割しサブファイルシステムに配置したサブファイル1A~1D、310~313は論理ファイル2を分割しサブファイルシステムに配置したサブファイル2A~2D、314~317はディスク装置AからD、318~321は二つのサブファイル、及びディスク装置を格納するサブファイルシステムA~Dを示す。サブファイルシステムAのディスク装置Aにおいて、論理ファイル1のストライドグループSG0と論理ファイル2のストライドグループSg0の書き込みが競合したとき、それぞれのファイルブロック群がストライドグループとして一括してディスク装置Aに書き込まれる。これにより、2つの異なる論理ファイルに属するファイルブロックが交互に（または、混じり合っ）てディスク装置に書き込まれることを避け、ディスク装置内の空き領域になるべく同一論理ファイルのファイルブロックが続けて配置されることで、ファイルブロックの読み出し時の回転待ち、シークを削減することが可能となり、ファイルの読み出し要求の効率化を図る。

【0011】図4に、サブファイルシステムにおいて、マスタファイルシステムより割り当てられたストライドグループをディスク装置に配置する処理の方法を説明する。S401で、マスタファイルシステムよりストライドグループ単位のファイルブロック群を受け取る。S402において、マスタファイルシステムから受け取ったファイルブロックを一括して、ディスク装置の空き領域に割り当てていく。本処理方式により、2論理ファイル以上のファイルブロック群を同一ディスク装置に書き込

もうとすると、必ず同一論理ファイルのファイルブロック群に対する書き込みが完了してから、次の論理ファイルのファイルブロック群に対する書き込みが開始されるため、異なる論理ファイルに属するファイルブロックが混在して、ディスク装置に書き込まれることが無いことを保証する。

【0012】次に本分散ファイル管理方法において、論理ファイルへの読み出し時のファイルアクセス方式について説明する。図5は、図1における論理ファイルに読み出し要求があった場合のファイルアクセス方法を説明するもので、501は論理ファイル、FB0～FB15は論理ファイルを一定のサイズに分割したファイルブロック、SG0～SG3はファイルブロックの集合であるストライドグループ、502～505はサブファイルAからD、506～509はディスク装置AからDを示す。図5で、論理ファイルのファイルブロック0から4に読み出し要求があった場合、ファイルブロック0から3までを並行に読み出すことが可能で、且つ、ディスク装置Aではファイルブロック0の読み出し要求を実行した後、直ちにファイルブロック4の読み出し要求を実行することが出来る。従来は、ファイルへの書き込み時、分割されたファイルブロック群に対して、同一ディスク装置へ格納すると認識せずに、各ブロックを別々にディスク装置に書き込んでいたため、ファイル読み出し時にディスク装置上でのファイルブロックの配置にばらつきが生じ、ディスク装置の回転待ち、シークの回数が増加し、性能劣化の原因となった。一方、同じ論理ファイルに属するファイルブロックを一纏めにして書き込んだ場合、読み出し時の回転待ち、シークの回数を削減し、効率よく同一論理ファイルのファイルブロック群を読み出すことが可能である。

【0013】次に示す実施例では、本ファイル管理方法において、サブファイルを読み出すとき、マスタファイルが管理している情報より、ファイルブロックを先読みし、それらをマスタファイルシステムのバッファキャッシュに保存する方法を示す。本ファイル管理方法では読み出し要求があった場合、最初にアクセスがあったファイルブロックより、同じディスク装置で次に読み出し要求が起り得るファイルブロックは現在読み出ししているファイルブロックと一括して書き込んでいるため、ファイルブロックを先読みして、マスタファイルシステムのバッファキャッシュに保存することが出来る。

【0014】図6は、本分散ファイル管理方法においてマスタファイルシステムにバッファキャッシュを装備したものを説明する図で、601はマスタファイル、602は構成定義ファイル、603は論理ファイル、604はバッファキャッシュ、FB0～FB15は論理ファイルを一定のサイズに分割したファイルブロック、SG0～SG3はファイルブロックの集合であるストライドグループ、605は601、602、603、及び604

を格納するマスタファイルシステム、606～609はサブファイルAからD、610～613はディスク装置AからD、614～617はサブファイル及びディスク装置を格納するサブファイルシステムAからDを示す。

【0015】バッファキャッシュは、マスタファイルシステムに装備し、サブファイルシステムより先読みしたファイルブロックを格納する領域として使用する。

【0016】図6において、サブファイルAのファイルブロック0にアクセスすると次にファイルブロック4、8、及び12にアクセスされることが起り得るため、これらの領域をファイルブロック0の読み出し要求が出た時点で、マスタファイルシステム上のバッファキャッシュに先読みすることで、再度マスタファイルシステムからサブファイルシステムへ読み出し要求を出すことなく、直接マスタファイルシステムが論理ファイルの読み出し要求に応えることが可能となる。このため、さらにファイルアクセスの高速化を図ることができる。

【0017】図7では、バッファキャッシュを装備したマスタファイルシステムを用い、読み出し要求があった場合のファイルアクセス方法を説明するもので、701は論理ファイル、702はバッファキャッシュ、FB0～FB15は論理ファイルを一定のサイズに分割したファイルブロック、SG0～SG3はファイルブロックの集合であるストライドグループ、703～706はサブファイルAからD、707～710はディスク装置AからDを示す。図7で、論理ファイルのファイルブロック0から8に読み出し要求があった場合、本方式では、ファイルブロック0に読み出し要求が発生すると、ファイルブロック0が属すストライドグループをバッファキャッシュにアップロードすることで、ファイルブロック4、8、12に読み出し要求する際にサブファイルシステムAのディスク装置Aにアクセスする必要がなくなり、代わりにバッファキャッシュにあるファイルブロックにアクセスすることが可能となる。同様にストライドグループSG1、SG2、及びSG3についてもバッファキャッシュにアップロードすることが出来る。このため、論理ファイルへの読み出し要求の範囲が大きいとき、ストライピングファイル機構を使って複数ディスク装置から並行してファイルブロックを読み出すとともに、1回のディスク装置への読み出し要求でディスク装置内に存在する後続のファイルブロックをアップロードできるため、以後サブファイルシステムに読み出し要求を出す必要がなくなり、ファイルアクセスの高速化を図ることが出来る。

【0018】図8、9に、本ファイルアクセス方法でバッファキャッシュを装備したときの処理について、フローチャートを用いて説明する。図8のフローチャートはマスタファイルシステムでの処理を、図9のフローチャートは各サブファイルシステムでの処理を示す。

【0019】図8の処理において、マスタファイルシス

テムは S801 でアプリケーションより読み出し要求を受けると、S802 の読み出し要求で、最初のファイルブロックにアクセスをする。このとき、S803 で、まず読み出し要求が発生したファイルブロックが、マスタファイルシステムのバッファキャッシュ内に存在するかを確認する。最初にアクセスしたファイルブロックは、キャッシュ内に存在しないため、S805 に移行する。S805 では、マスタファイルが管理している情報より、配置したサブファイルシステムに対して、当該ファイルブロックと後続するファイルブロックの読み出し要求を出す。2 つ目以降のファイルブロックのアクセスでは、当該ファイルブロックがバッファキャッシュに格納されている場合もあるため、その場合は S803 から S804 に移行する。S804 では、マスタファイルシステムのバッファキャッシュより読み出し要求のあったファイルブロックを読み出す。S806 で、読み出し要求のあった全てのファイルブロックへのアクセスが完了したかを確認する。読み出し要求への読み出しが完了している場合、処理を終了する。読み出し要求に対する読み出しが完了していない場合、S807 で読み出し要求の出ている次のファイルブロックにアクセスをする。このとき、ファイルブロックが、バッファキャッシュ内に存在しているかを調べるため、S803 に戻る。このように、マスタファイルシステムでは、順次読み出し要求のあったファイルブロックにアクセスをして、バッファキャッシュに当該ファイルブロックが存在する場合、バッファキャッシュのファイルブロックを用い読み出し要求に応え、バッファキャッシュに無い場合、当該ファイルブロックを配置しているサブファイルシステムに読み出し要求を出す。

【0020】図 9 で、バッファキャッシュを装備する方法において、サブファイルシステム側の処理を説明する。まず、S901 でマスタファイルシステムから読み出し要求を受け取り、S902 で読み出し要求のあったファイルブロックから読み出し、マスタファイルシステムにファイルブロックを送信する。S903 において、読み出し要求のあった同一ストライドグループのファイルブロックがあると、そのブロックに対しての読み出すため S902 の処理を繰り返す。S903 において、同一ストライドグループに属する全てのファイルブロックの読み出しが完了していると処理が終了する。このとき、本方法では、ファイル書き込み時にストライドグループを用い、同一論理ファイルのファイルブロックを一括して当該ディスク装置の連続した領域に配置しているため、読み出し要求時、ファイルブロックを一括して読み出すことが可能となる。

【0021】

【発明の効果】本発明では、複数の論理ファイルのファイルブロックが同一ディスク装置に配置される場合、これらのファイルブロックをディスク装置内で混在して配置することを回避し、それぞれのファイルブロックを一括して書き込むことで、論理ファイルの読み出し時、読み出し要求のあったファイルブロックを検索する時間の短縮を可能にする。また、ファイルブロックの先読みをすることで、マスタファイルシステムからサブファイルシステムへの読み出し要求の回数を削減することができ、大容量ファイルアクセス時間の短縮を実現する。

【図面の簡単な説明】

【図 1】本発明の一実施例のマスタファイルシステムとサブファイルシステムの構成を示す全体図である。

【図 2】図 1 の実施例において、論理ファイルの分割方法の処理を示すフローチャートである。

【図 3】図 1 の実施例において、複数の論理ファイルが同一サブファイルシステムを共有したときを示す図である。

【図 4】図 1 の実施例において、書き込みに対してサブファイルシステム側でのファイルブロックの書き込み処理を示すフローチャートである。

【図 5】図 1 の実施例において、ストライドグループがそれぞれのサブファイルシステムに分割された後、論理ファイルに読み出し要求があったときの、ファイルアクセス方法を示す図である。

【図 6】本発明の実施例である分散ファイル管理方式で、マスタファイルシステムにバッファキャッシュを装備して、ファイル読み出しのあったファイルブロックと同一のストライドグループに属するファイルブロックを、マスタファイルシステムに先読みする方法を示す全体構成図である。

【図 7】図 6 のバッファキャッシュを装備する実施例で、論理ファイルに読み出し要求があったときの、ファイルアクセス方法とファイルブロックの先読み方法の処理を示す図である。

【図 8】図 6 のファイルブロックを先読みする実施例で、マスタファイルシステムに読み出し要求があった場合の処理を示すフローチャートである。

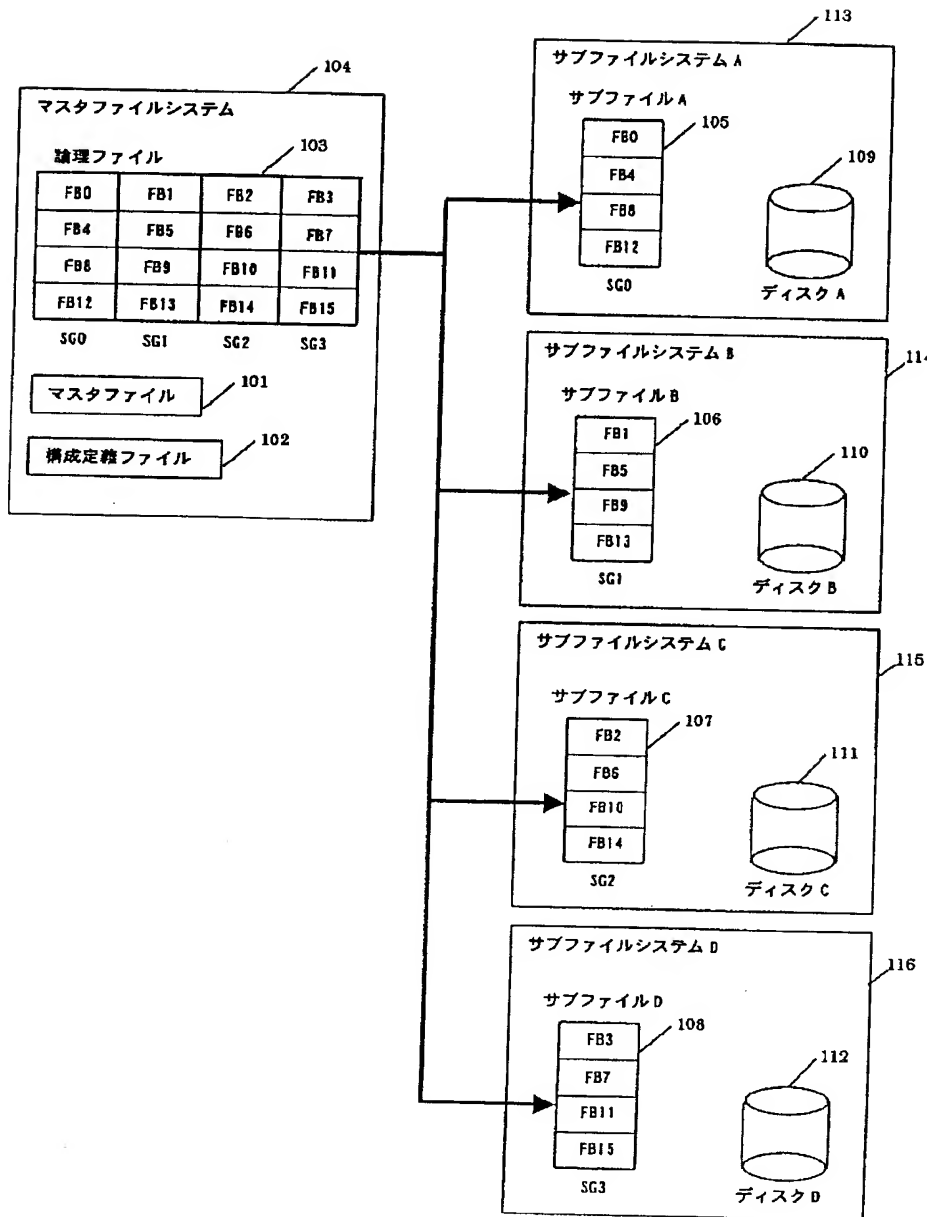
【図 9】図 6 のファイルブロックを先読みする実施例で、サブファイルシステムがマスタファイルシステムから読み出し要求を受けたときの処理を示すフローチャートである。

【符号の説明】

602…構成定義ファイル、603…論理ファイル、604…バッファキャッシュ、605…マスタファイルシステム、606…サブファイル A、610…ディスク装置 A、614…サブファイルシステム A。

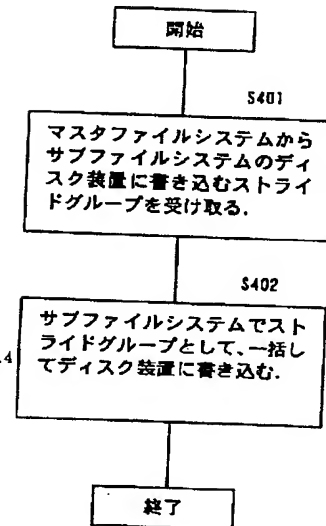
【図1】

図 1

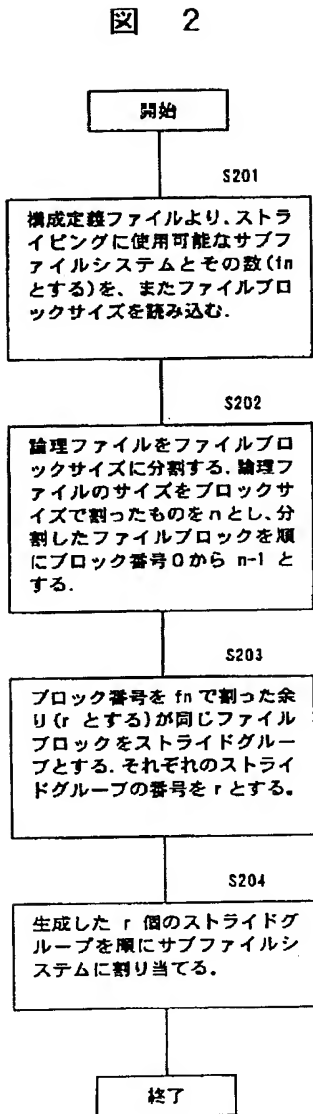


【図4】

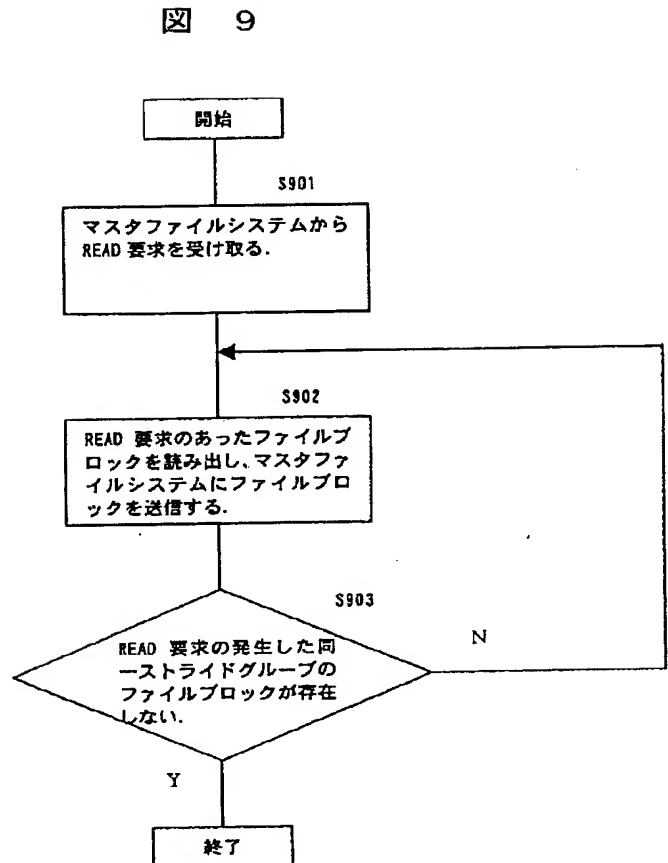
図 4



【図 2】

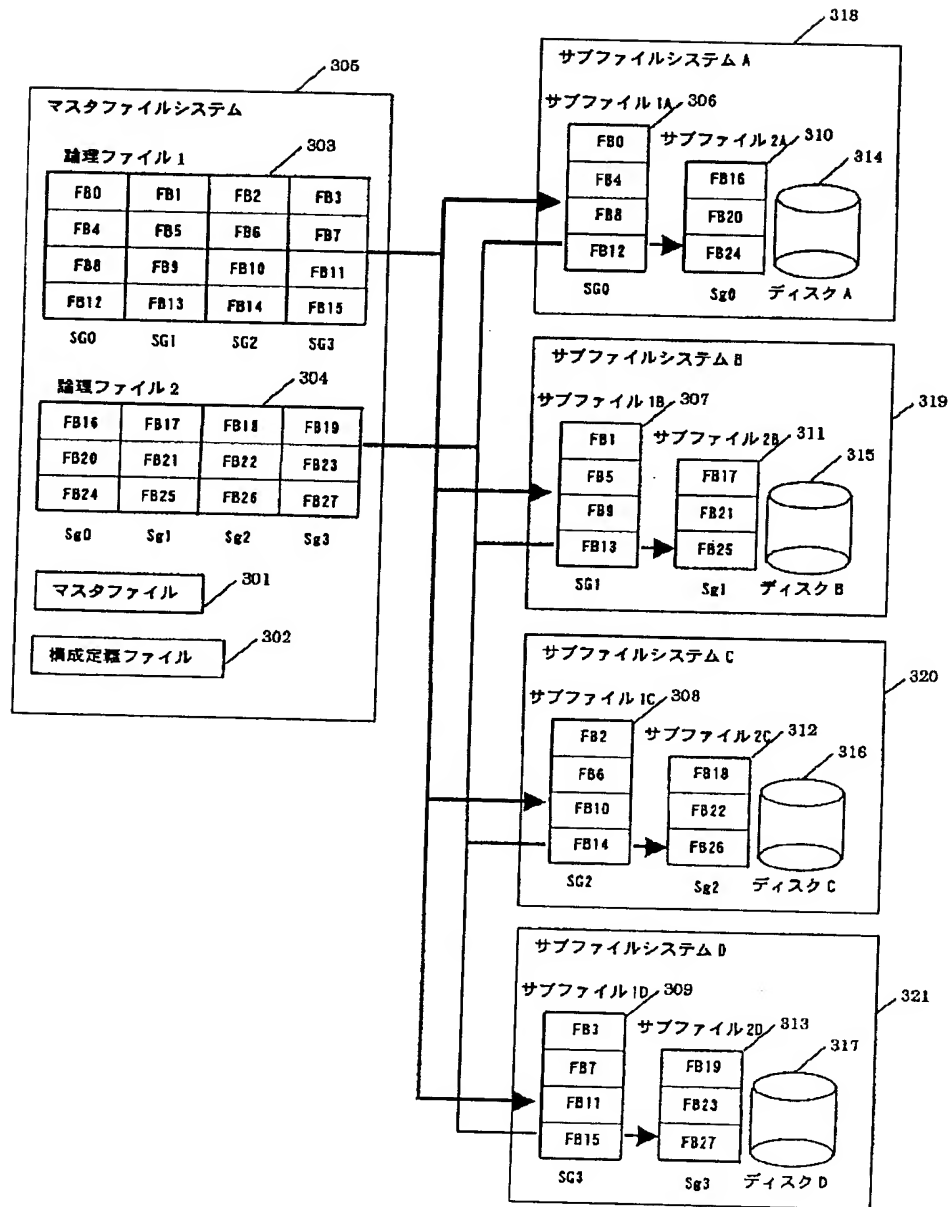


【図 9】



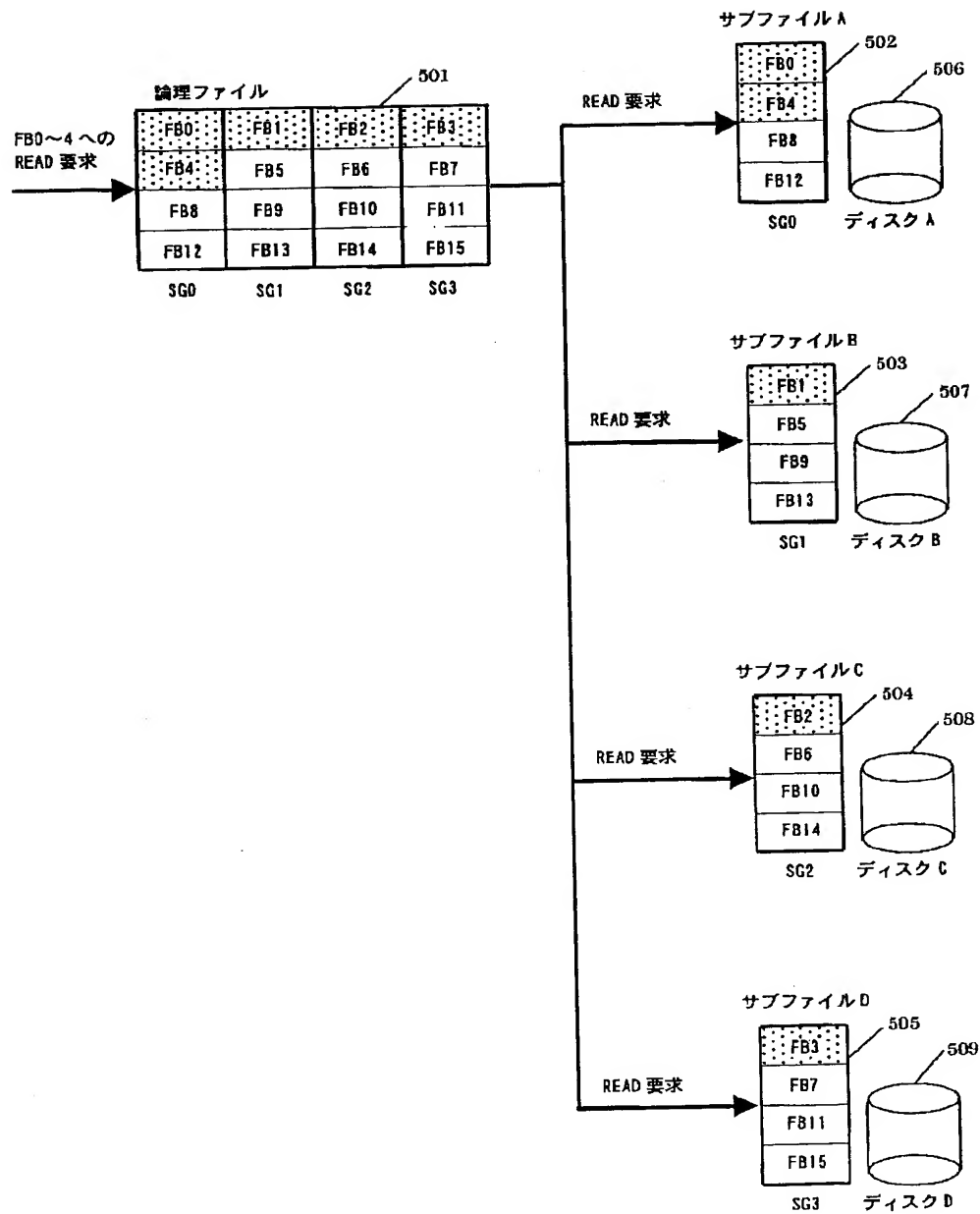
【図 3】

図 3



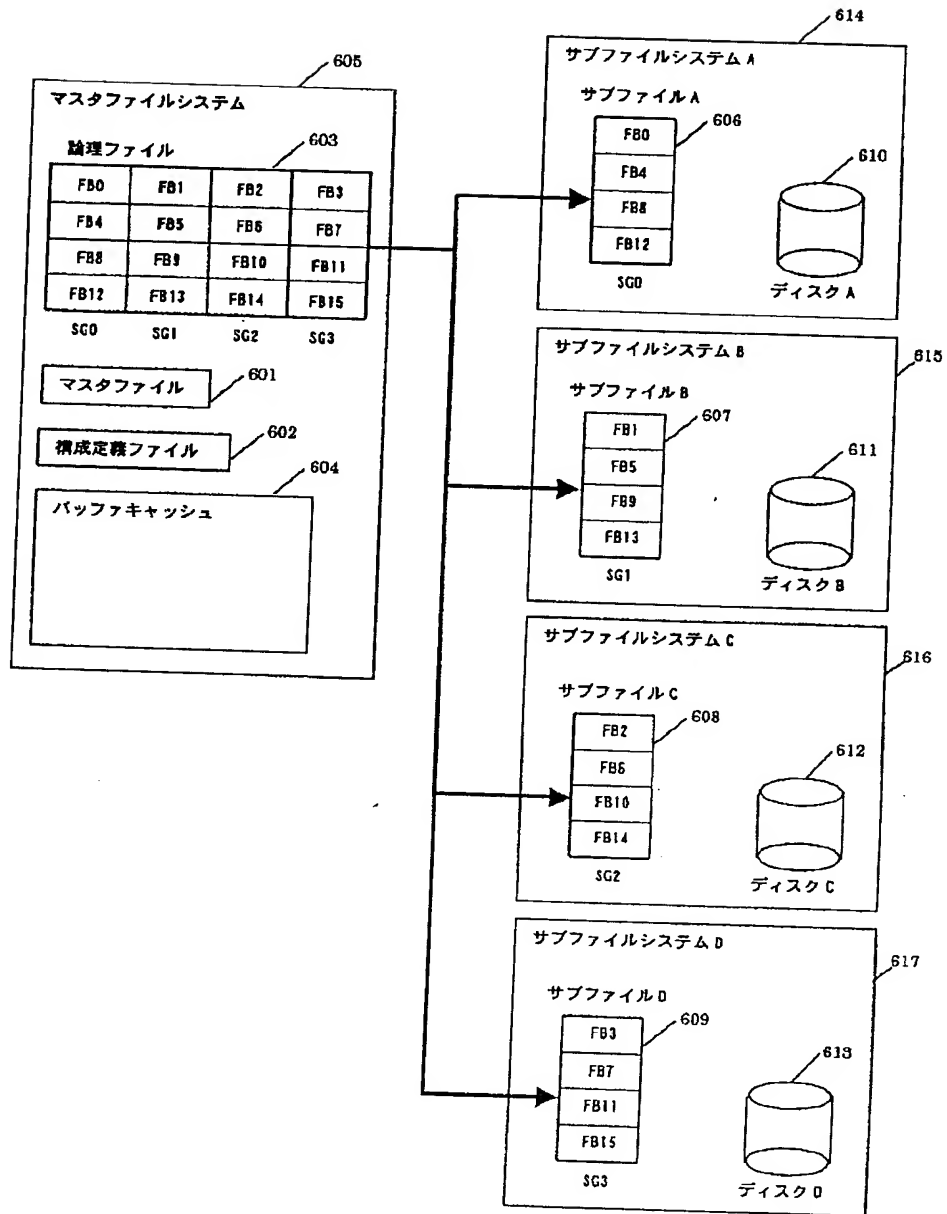
【図 5】

図 5



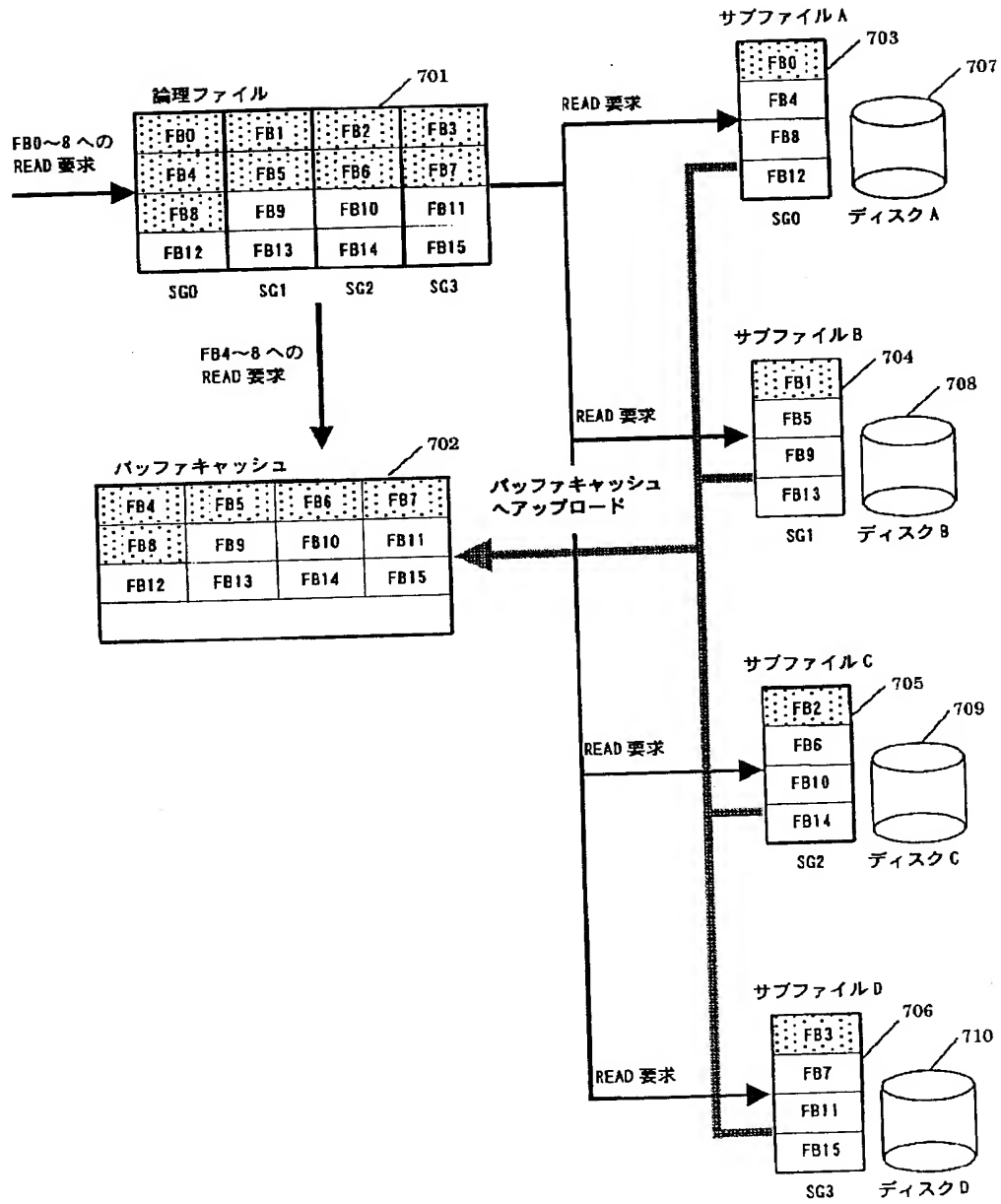
【図 6】

図 6



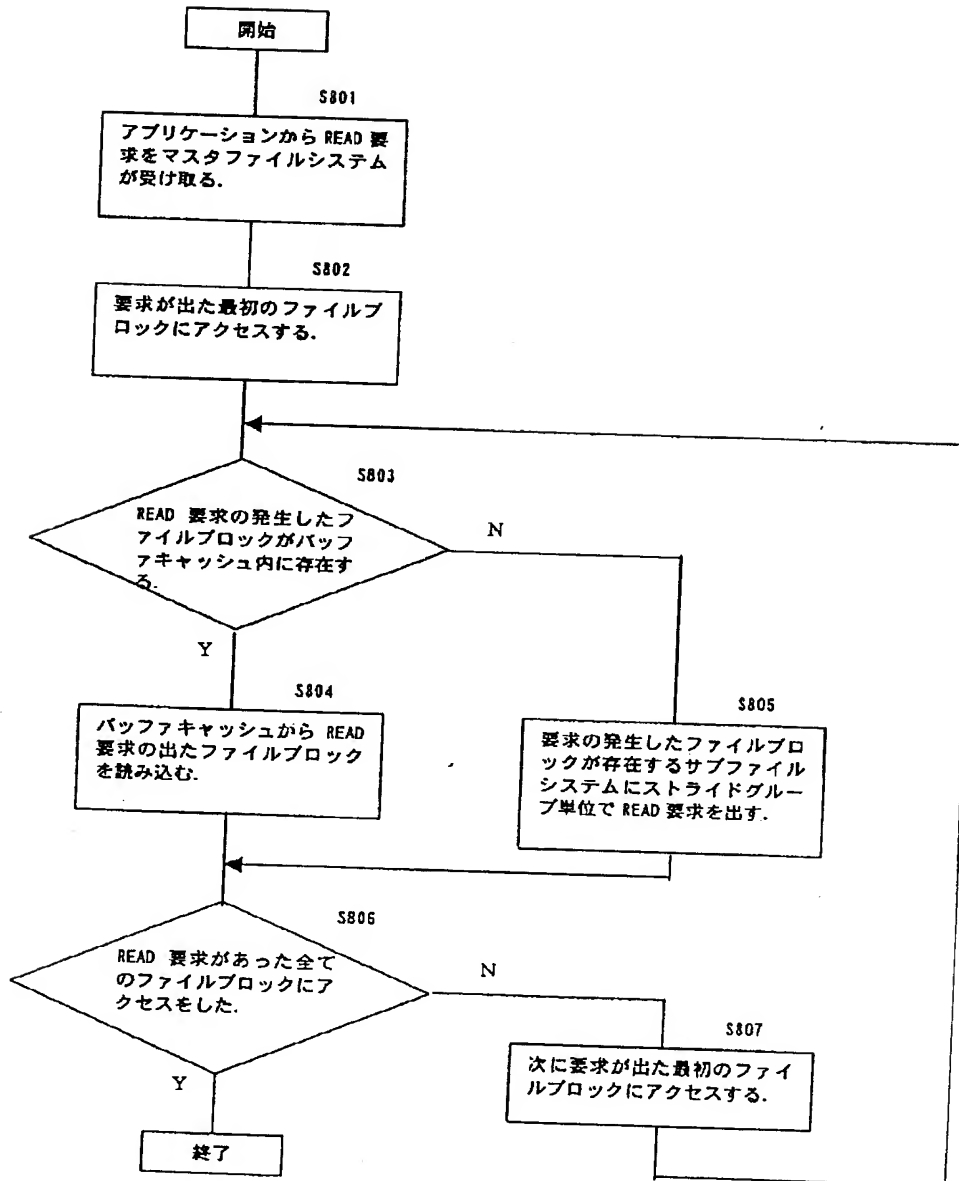
【図 7】

図 7



【図 8】

図 8



フロントページの続き

(72) 発明者 杉山 和仁
 神奈川県横浜市戸塚区戸塚町5030番地 株
 式会社日立製作所ソフトウェア開発本部内

(72) 発明者 松井 和吉
 神奈川県横浜市中区尾上町六丁目81番地
 日立ソフトウェアエンジニアリング株式会
 社内

F ターム(参考) 5B065 BA01 CH02
5B082 CA18 CA20 FA12
5D110 AA13 DA11 DB05 DC03 DC16
DE01

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-182953

(43)Date of publication of application : 28.06.2002

(51)Int.Cl.

G06F 12/00

G06F 3/06

G11B 27/00

(21)Application number : 2000-383124

(71)Applicant : HITACHI LTD

HITACHI SOFTWARE ENG CO LTD

(22)Date of filing : 12.12.2000

(72)Inventor : NISHIMURA YUJI

SUGIYAMA KAZUHITO

MATSUI WAKICHI

(54) DISTRIBUTED FILE MANAGEMENT METHOD

(57)Abstract:

PROBLEM TO BE SOLVED: To accelerate parallel disk device access at the time of I/O requests in a method of distributing and arranging a plurality of logic files to disk units by a striping file mechanism.

SOLUTION: In this method of distributing and arranging the logic file 603 to sub-filing systems 614-617, stride groups SG0-SG3 which are the sets of file blocks are generated so as to arrange the file blocks FB0-FB15 of the same file in the batch area of the disk devices 610-613 and they are written altogether to the respective sub-filing systems by the stride group unit. Also, when accessing the file block, the successive file blocks FB4-FB15 are looked ahead to the buffer cache 604 of a master filing system 605, the number of times of requesting read to the sub-filing systems is reduced and file access is accelerated further.

